COMPUTER SCIENCE

A.K. Watson Hall, 203.432.1246
http://cpsc.yale.edu
M.S., M.Phil., Ph.D.
Chair
Zhong Shao

Director of Graduate Studies
Vladimir Rokhlin (108 AKW, 203.432.1278, vladimir.rokhlin@yale.edu)


Senior Lecturers James Glenn, Kyle Jensen, * Stephen Slade

Lecturers Timothy Barron, Andrew Bridy, † Rob Brunstad, Jay Lim, Cody Murphey, Scott Petersen, Brad Rosen, Andrew Sherman, * Inyoung Shin, Cecillia Xie

* A secondary appointment with primary affiliation in another department or school.
† A joint appointment with another department.

FIELDS OF STUDY
Algorithms and computational complexity, artificial intelligence, data networking, databases, graphics, machine learning, programming languages, robotics, scientific computing, security and privacy, and systems.

RESEARCH FACILITIES
The department operates a high-bandwidth, local-area computer network based mainly on distributed workstations and servers, with connections to worldwide networks. Workstations include Dell dual-processor PCs (running Linux or Windows/XP). Laboratory contains specialized equipment for graphics, vision, and robotics research. Various printers, including color printers, as well as image scanners, are also available. The primary educational facility consists of thirty-seven PC workstations supported by a large Intel PC server. This facility is used for courses and unsponsored research by Computer Science majors and first-year graduate students. Access to computing, through both the workstations and remote login facilities, is available to everyone in the department.

SPECIAL REQUIREMENTS FOR THE PH.D. DEGREE
There is no foreign language requirement. To be admitted to candidacy, a student must (1) pass ten courses (including CPSC 690 and CPSC 691) with at least two grades of Honors, the remainder at least High Pass, including three advanced courses in an area of specialization; (2) take six advanced courses in areas of general computer science; (3) successfully complete a research project in CPSC 690, CPSC 691, and submit a written report on it to the faculty; (4) pass a qualifying examination in an area of specialization; (5) be accepted as a thesis student by a regular department faculty member; (6) serve as a teaching assistant for two terms; and (7) submit a written dissertation prospectus, with a tentative title for the dissertation. To satisfy the distribution requirement (requirement 2 above), the student must take one course in programming languages or systems, one programming-intensive course, two theory courses, and two in application areas. In order to gain teaching experience, all graduate students are required to serve as teaching assistants for two terms during their first three years of study. All requirements for admission to candidacy must be completed prior to the end of the third year. In addition to all other requirements, students must successfully complete CPSC 991, Ethical Conduct of Research, prior to the end of their first year of study. This requirement must be met prior to registering for a second year of study.

MASTER’S DEGREES
M.Phil. See Degree Requirements under Policies and Regulations.

M.S. (en route to the Ph.D.) To qualify for the M.S., the student must pass eight courses at the 500 level or above from an approved list. An average grade of at least High Pass is required, with at least one grade of Honors.
Improving their services by analyzing the collected data. This introduces interesting new opportunities and challenges designing systems.

Today's Internet-scale applications and cloud services generate massive amounts of data. At the same time, the availability of inexpensive storage has made it possible for these services and applications to collect and store every piece of data they generate, in the hopes of improving their services by analyzing the collected data. This introduces interesting new opportunities and challenges designing systems.
for collecting, analyzing, and serving the so-called big data. This course looks at technology trends that have paved the way for big data applications, surveys state-of-the-art systems for storage and processing of big data, and considers future research directions driven by open research problems. Our discussions span topics such as cluster architecture, big data analytics stacks, scheduling and resource management, batch and stream analytics, graph processing, ML/AI frameworks, and serverless platforms and disaggregated architectures.

**CPSC 530b, Software Engineering**  
Timos Antonopoulos  
Introduction to building a large software system in a team. Learning how to collect requirements and write a specification. Project planning and system design. Increasing software reliability: debugging, automatic test generation. Introduction to type systems, static analysis, and model checking.

**CPSC 546a, Data and Information Visualization**  
Holly Rushmeier  
Visualization is a powerful tool for understanding data and concepts. This course provides an introduction to the concepts needed to build new visualization systems, rather than to use existing visualization software. Major topics are abstracting visualization tasks, using visual channels, spatial arrangements of data, navigation in visualization systems, using multiple views, and filtering and aggregating data. Case studies to be considered include a wide range of visualization types and applications in humanities, engineering, science, and social science. Prerequisite: CPSC 223.

**CPSC 540b / ENAS 907b, Computer Architectures and Artificial Intelligence**  
Richard Lethin  
Introduction to the development of computer architectures specialized for cognitive processing, both offline “thinking machines” as well as embedded devices. History of machines starting with early conceptions in defense systems to contemporary initiatives. Instruction sets, memory systems, parallel processing, analog architectures, probabilistic architectures, graph computing architectures, machine-learning architectures. Application and algorithm characteristics.

**CPSC 551b, The User Interface**  
David Gelernter  
The user interface (UI) in the context of modern design, where tech has been a strong and consistent influence from the Bauhaus and U.S. industrial design of the 1920s and 1930s through the IBM-Eames design project of the 1950s to 1970s. The UI in the context of the windows-menus-mouse desktop, as developed by Alan Kay and Xerox in the 1970s and refined by Apple in the early 1980s. Students develop a detailed design and simple implementation for a UI.

**CPSC 552b / AMTH 552b / CB&B 663b, Deep Learning Theory and Applications**  
Smita Krishnaswamy  
Deep neural networks have gained immense popularity within the past decade due to their success in many important machine-learning tasks such as image recognition, speech recognition, and natural language processing. This course provides a principled and hands-on approach to deep learning with neural networks. Students master the principles and practices underlying neural networks, including modern methods of deep learning, and apply deep learning methods to real-world problems including image recognition, natural language processing, and biomedical applications. Course work includes homework, a final exam, and a final project—either group or individual, depending on enrollment—with both a written and oral (i.e., presentation) component. The course assumes basic prior knowledge in linear algebra and probability. Prerequisites: CPSC 202 and knowledge of Python programming.

**CPSC 553a / AMTH 553a / CB&B 555a / GENE 555a, Unsupervised Learning for Big Data**  
Smita Krishnaswamy  
This course focuses on machine-learning methods well-suited to tackling problems associated with analyzing high-dimensional, high-throughput noisy data including: manifold learning, graph signal processing, nonlinear dimensionality reduction, clustering, and information theory. Though the class goes over some biomedical applications, such methods can be applied in any field. Prerequisites: knowledge of linear algebra and Python programming.

**CPSC 554b, Software Analysis and Verification**  
Ruzica Piskac  
Introduction to concepts, tools, and techniques used in the formal verification of software. State-of-the-art tools used for program verification; detailed insights into algorithms and paradigms on which those tools are based, including model checking, abstract interpretation, decision procedures, and SMT solvers.

**CPSC 555a / ENAS 951a, Wireless Technologies and the Internet of Things**  
Wenjun Hu  
Fundamental theory of wireless communications and its application explored against the backdrop of everyday wireless technologies such as WiFi and cellular networks. Channel fading, MIMO communication, space-time coding, opportunistic communication, OFDM and CDMA, and the evolution and improvement of technologies over time. Emphasis on the interplay between concepts and their implementation in real systems. The labs and homework assignments require Linux and MATLAB skills and simple statistical and matrix analysis (using built-in MATLAB functions).

**CPSC 557a, Sensitive Information in a Connected World**  
Michael Fischer  
Issues of ownership, control, privacy, and accuracy of the huge amount of sensitive information about people and organizations that is collected, stored, and used by today’s ubiquitous information systems. Readings consist of research papers that explore both the power and the limitations of existing privacy-enhancing technologies such as encryption and “trusted platforms.”

**CPSC 558b, Automated Decision Systems**  
Stephen Slade  
People make dozens of decisions every day in their personal and professional lives. What would it mean for you to trust a computer to make those decisions for you? It is likely that many of those decisions are already informed, mediated, or even made by computer systems. Explicit examples include dating sites like match.com or recommendation systems such as Amazon or Netflix. Most Internet ads on sites like Google or Facebook are run by real-time-bidding (RTB) systems that conduct split-second auctions in the hopes of
Students complete a final project, which can be a synthesis review of recent development and state-of-the-art results in some machine-domains. The course provides a foundation for students interested in pursuing further research or applications of machine learning. This course provides an introduction to machine learning and the problem of learning from data. It introduces several frameworks to enable interactive machines to perceive and act in a variety of environments. Part of the course examines approaches for perception with different sensing devices and algorithms; the other part focuses on methods for decision-making and applied machine learning for control. The course is a combination of lectures, state-of-the-art reading, presentations and discussions, programming assignments, and a final team project. Prerequisites: CPSC 570 and understanding of probability, differential calculus, linear algebra, and planning (in Artificial Intelligence). Programming assignments require proficiency in Python and high-level familiarity with C++. Students who do not fit this profile may be allowed to enroll with the permission of the instructor.

CPSC 560a, Automata Theory and Formal Languages  Andrew Bridy
Introduction to the theory of automata and formal languages, one of the building blocks of theoretical computer science. Major topics covered are finite automata, pushdown automata, and Turing machines, and their associated languages.

CPSC 562a / ECON 565a, Algorithms via Continuous Optimization  Nisheeth Vishnoi
Continuous optimization has played a major role in the recent development of fast algorithms for problems arising in areas such as theoretical computer science, discrete optimization, and machine learning. The approach is to first formulate the problem as a continuous optimization problem, even if the problem may be over a discrete domain; adapt or develop deterministic or randomized continuous-time dynamical systems to solve it; and then design algorithms for the problem via appropriate discretizations. The goal of this course is to design state-of-the-art algorithms for various classical discrete problems through the use of continuous optimization/sampling. The algorithmic applications include shortest paths, bipartite matching, flows, linear programming, sampling, and counting. We present approaches including gradient descent, mirror descent, multiplicative weights update method, accelerated gradient descent, Riemannian descent, Newton’s method, cutting-plane methods, Langevin dynamics, and Hamiltonian dynamics. Prerequisite: CPSC 365 or CPSC 366 or permission of the instructor. S&DS 630 and a solid background in calculus, linear algebra, probability, and algorithms are recommended.

CPSC 570b, Artificial Intelligence  Brian Scassellati
Introduction to artificial intelligence research, focusing on reasoning and perception. Topics include knowledge representation, predicate calculus, temporal reasoning, vision, robotics, planning, and learning.

CPSC 572a, Intelligent Robotics  Brian Scassellati
Introduction to the construction of intelligent, autonomous systems. Sensory-motor coordination and task-based perception. Implementation techniques for behavior selection and arbitration, including behavior-based design, evolutionary design, dynamical systems, and hybrid deliberative-reactive systems. Situated learning and adaptive behavior.

CPSC 574a, Computational Intelligence for Games  James Glenn
An overview of computational vision with a biological emphasis. Suitable as an introduction to biological perception for computer science and engineering students, as well as an introduction to computational vision for mathematics, psychology, and physiology students.

CPSC 577b, Natural Language Processing  Staff
Linguistic, mathematical, and computational fundamentals of natural language processing (NLP). Topics include part of speech tagging, Hidden Markov models, syntax and parsing, lexical semantics, compositional semantics, machine translation, text classification, discourse, and dialogue processing. Additional topics such as sentiment analysis, text generation, and deep learning for NLP.

CPSC 578b, Computer Graphics  Julie Dorsey
Introduction to the basic concepts of two- and three-dimensional computer graphics. Topics include affine and projective transformations, clipping and windowing, visual perception, scene modeling and animation, algorithms for visible surface determination, reflection models, illumination algorithms, and color theory.

CPSC 579a, Advanced Topics in Computer Graphics  Julie Dorsey
An in-depth study of advanced algorithms and systems for rendering, modeling, and animation in computer graphics. Topics vary and may include reflectance modeling, global illumination, subdivision surfaces, NURBS, physically based fluids systems, and character animation.

CPSC 581a, Introduction to Machine Learning  Andre Wibisono
This course provides an introduction to machine learning and the problem of learning from data. It introduces several frameworks for formulating the learning task as statistical and computational problems, and explores algorithms for solving them. Topics include supervised learning (classification, regression, kernel methods, neural networks), unsupervised learning (clustering, PCA, dimensionality reduction), reinforcement learning (Markov decision process, online learning), and examples of machine-learning applications in various domains. The course provides a foundation for students interested in pursuing further research or applications of machine learning. Students complete a final project, which can be a synthesis review of recent development and state-of-the-art results in some machine-
learning applications. It should also have a research component, for example exploring different algorithms or generalizing the results to different applications, ideally related to each student's own research.

**CPSC 610a, Topics in Computer Science and Law**  Joan Feigenbaum  
This course focuses on socio-technical problems in computing, i.e., problems that cannot be solved through technological progress alone but rather require legal, political, or cultural progress as well. Examples include but are not limited to computer security, intellectual property protection, cyber crime, cyber war, surveillance, and online privacy. The course is addressed to graduate students in Computer Science who are interested in socio-technical issues but whose undergraduate work may not have addressed them; it is designed to bring these students rapidly to the point at which they can do research on socio-technical problems. Students do term projects (either papers or software artifacts) and present them at the end of the term. In order to ensure that there is enough time for both midterm feedback on project proposals and in-class presentation of the finished projects, enrollment is limited to fifteen. If fewer than fifteen Computer Science graduate students enroll, Yale College undergraduates will be allowed to enroll with permission of the instructor. Prerequisites: the basics of cryptography and computer security (as covered in CPSC 467), networks (as covered in CPSC 433), and databases (as covered in CPSC 437), or permission of the instructor.

**CPSC 640b / AMTH 640b, Topics in Numerical Computation** Staff  
This course discusses several areas of numerical computing that often cause difficulties to non-numericists, from the ever-present issue of condition numbers and ill-posedness to the algorithms of numerical linear algebra to the reliability of numerical software. The course also provides a brief introduction to “fast” algorithms and their interactions with modern hardware environments. The course is addressed to Computer Science graduate students who do not necessarily specialize in numerical computation; it assumes the understanding of calculus and linear algebra and familiarity with (or willingness to learn) either C or FORTRAN. Its purpose is to prepare students for using elementary numerical techniques when and if the need arises.

**CPSC 647a or b, Quantum Computer Systems** Staff  
This course explores advanced topics in quantum computer systems design. We spend some time developing the mathematical basis of quantum computation and information, then examine powerful techniques in quantum application and architecture design. Students learn to work with IBM Qiskit software tools to write quantum programs and execute them on cloud-accessible quantum hardware. Topics covered include quantum programming, quantum algorithms, quantum compilation, memory management, quantum error correction, and some research topics. Students are asked to read and discuss selected research papers and conduct novel research in a term-long individual/group project. We anticipate the course will be of interest to students working in computer science, electrical engineering, physics, and mathematics. It is intended primarily for Ph.D. and M.S. students; advanced undergraduates may enroll with the permission of the instructor. Prerequisites: students must be comfortable with discrete probability, linear algebra, and computer architecture (e.g., CPSC 323). Prior experience in quantum computing is useful but not required.

**CPSC 674b, Advanced Computational Intelligence for Games** James Glenn  
A seminar on current topics in computational intelligence for games, including developing agents for playing games, procedural content generation, and player modeling. Students read, present, and discuss recent papers and competitions, and complete a term-long project that applies some of the techniques discussed during the term to a game of their choice. Prerequisite: CPSC 574.

**CPSC 677a, Advanced Natural Language Processing** Dragomir Radev  
Advanced topics in natural language processing (NLP), including related topics such as deep learning and information retrieval. Included are: (1) fundamental material not covered in the introductory NLP class such as text summarization, question answering, document indexing and retrieval, query expansion, graph-based techniques for NLP and IR, as well as (2) state-of-the-art material published in the past few years such as transfer learning, generative adversarial networks, reinforcement learning for NLP, sentence representations, capsule networks, multitask learning, and zero-shot learning. Prerequisite: CPSC 570, CPSC 577, or equivalent, or permission of the instructor.

**CPSC 690a, Independent Project I** Staff  
By arrangement with faculty.

**CPSC 691a, Independent Project II** Staff  
By arrangement with faculty.

**CPSC 752b / CB&B 752b / MB&B 753b and MB&B 754b / MB&B 753b and MB&B 754b / MB&B 754b / MCDB 752b, Biomedical Data Science: Mining and Modeling** Mark Gerstein and Matthew Simon  
Biomedical data science encompasses the analysis of gene sequences, macromolecular structures, and functional genomics data on a large scale. It represents a major practical application for modern techniques in data mining and simulation. Specific topics to be covered include sequence alignment, large-scale processing, next-generation sequencing data, comparative genomics, phylogenetics, biological database design, geometric analysis of protein structure, molecular-dynamics simulation, biological networks, normalization of microarray data, mining of functional genomics data sets, and machine-learning approaches to data integration. Prerequisites: biochemistry and calculus, or permission of the instructor.

**CPSC 900a or b, Ethical Conduct of Research for Master's Students** Holly Rushmeier  
This course meets on four consecutive Fridays.

**CPSC 991a / MATH 991a, Ethical Conduct of Research** Holly Rushmeier  